基于混合人工蜂群算法的多目标柔性作业车间调度问题研究 *

孟冠军，杨大春，陶细佩

(合肥工业大学 机械工程学院，合肥 230009)

摘　要：传统的优化算法在求解面对多目标柔性作业车间调度时，往往求解效率低且难以获得最优解。为了求解多目标柔性作业车间调度问题，设计了混合人工蜂群算法。种群的初始化采用了多种方法相结合的策略。在人工蜂群算法的不同阶段采用不同的搜索机制，在雇佣蜂阶段采用开发搜索，针对跟随蜂阶段蜜蜂跟随的对象的优秀解进行小幅度的更新，从而提高了搜索的表现。禁忌搜索与改进的人工蜂群算法相结合，有效的提升了获得最优解的概率。通过相关文献中的标准实例对设计的混合人工蜂群算法进行一系列求解测试，实验的结果有效的说明了算法在求解柔性作业车间调度问题时效果显著。通过求解结果对比表明人工蜂群算法的高效性和优越性。

关键词：计算机应用；柔性作业车间调度；人工蜂群算法；多目标优化；禁忌搜索

中图分类号：TP399

Study on multi-objective flexible Job-Shop scheduling problem based on hybrid artificial bee colony algorithm

Meng Guanjun, Yang Dachun, Tao Xipei

(*School of mechanical engineering*, *Hefei university of technology*, *Hefei* 230009, *China*)

Abstract: When solving multi-objective flexible job shop scheduling problem, the traditional optimization algorithms were often inefficient and difficult to obtain the optimal solution. In order to solve the multi-objective flexible job shop scheduling problem, a hybrid artificial bee colony algorithm was designed. The method in population initialization phase was combination of several strategies. Different search mechanisms are employed at different stages of the artificial bee colony algorithm: the traditional exploit search was adopted in the employed bee phase, and objects of bees following were updated with little range in onlooker bee phase. The combination of tabu search and improved artificial bee colony algorithm effectively improved the probability of obtaining the optimal solution. The paper tested hybrid artificial bee colony algorithm through the Kacem instances. The results showed that the hybrid artificial bee colony algorithm is efficient in solving flexible job shop scheduling problem. The results indicated the efficiency and superiority of artificial bee colony algorithm.

Key Words: computer application; Flexible Job-Shop scheduling problem; artificial bee colony algorithm; Multi-objective optimization; tabu search

0　引言

柔性作业车间调度(flexible Job-Shop scheduling problem,FJSP)是作业车间调度的扩展，一道工序可选生产设备有多个，需要从中选择一个最合适的。在实际生产过程中，需要考虑两个问题：设备分配问题，为工序选择一个合适的生产设备；工序排列问题，对众多的工序操作进行合理的安排。因此 FJSP 比 JSP 复杂，更加符合生产实际，调度时更加灵活，是当前众多企业需要解决的问题[1]。

最近这些年，很多专家以元启发式算法求解 FJSP 问题。张国辉等人[2]建立了基于工件移动时间的柔性作业车间调度模型。Wang 等人[3]通过基于 FBS 的算法求解机器保养工作的 FJSP 问题。苏子林等人[4]设计了基于组合优先规则的启发式算法求解 FJSP。Shao 等人[5]提出 DPSO 和 SA 混合算法求解多目标 FJSP，PSO 和 SA 算法分别用来指导全局搜索和局部搜索从而使算法的性能有所改善。Zhang 等人[6]提出了基于免疫和熵原理的遗传算法求解多目标 FJSP。免疫和熵原理用来保持个体的多样性克服过早收敛问题。马雪丽等[7]针对柔性工艺的 FJSP，提出了混合遗传算法。Zhang 等人[8]发展了混合粒子群算法求解 FJSP，其中 PSO 与禁忌搜索相结合用来求解多个相互冲突的相差悬殊的目标的多目标问题。吴秀丽等人[9]针对柔性作业车间调度问题的 NP 难特性，提出一种改进的细菌觅食优化算法。

基金项目：马鞍山市科技计划资助项目

作者简介：孟冠军（1979-），男，安徽临泉人，副教授，博士，主要研究方向为数字化设计与制造、制造业信息化工程(mgj1997@163.com)；杨大春（1991-），男，硕士研究生，主要研究方向为计算机综合自动化；陶细佩（1993-），男，硕士研究生，主要研究方向为制造系统检测、控制与管理.

在求解 FJSP 方面，人工蜂群算法(artificial bee colony algorithm, ABC)具有速度快、效率高等优点。Li 等[10]提出了基于 Pareto 的混合 ABC 算法求解多目标 FJSP。Thammano[11]设计了局部搜索的混合 ABC 算法求解 FJSP。Banharnsakun[12]提出了基于 Best-so-far 的人工蜂群算法求解 FJSP。Wang 等人[13]使用多项策略产生初始化解，使用 Pareto 存储和记录非支配解。

为了提高 ABC 算法求解 FJSP 获得最优解的能力，本文使用了开发搜索和新的搜索机制，解的初始化方法也得到改进。禁忌搜索(tabu search, TS)允许接受差解，具有较强的局部搜索能力。本文将 TS 与 ABC 算法相结合，提出了混合人工蜂群算法(HTABC)，通过实验证明了算法的求解速度和解的质量都得到提高。

1　FJSP 模型

在介绍 FJSP 模型之前，首先定义几个重要的符号：

m 和 n 分别表示机器和工件；

$O_{i,h}$ 表示工件 i 的第 h 个工序；

n_i 表示工件 i 的工序总数；

C_{MAX} 为最大完工时间 makespan；

W_l 最大机器负荷；

W_{al} 机器总负荷；

C_i 表示工序 i 的完工时间；

P_{ijk} 表示工件 i 的第 j 个工序使用机器 k 生产的操作时间；

x_{ijk} 表示工件 i 的第 j 个工序是否使用机器 k 生产，如果是，则 x_{ijk} 为 1，否则 x_{ijk} 为 0。

在实际机器环境下，FJSP 可以这样被描述：给定的 n 个工件由 m 个机器生产，每个工件都由若干工序组成，每个工序都可以选择若干生产设备，工件在不同的设备上生产所花费的时间是不一样的。因此调度的目的就是在设备能力约束和工件约束条件下，确定最佳的加工顺序和合理地选择生产设备。

假设条件：

a）所有机器在刚开始均处于空闲状态且可正常使用；

b）一台设备同时只能生产一个工件；

c）工件一旦开始生产便不能中断；

d）不同工件之间的优先级相同，不存在先后约束；

e）同一机器上的加工任务完成后才能开始另一项任务。

针对实际生产的需要，本文考虑 3 种性能指标：最大完工时间 C_{max}，最大机器负荷 W_l 以及机器总负荷 W_{al}。

本文需要同时进行优化的目标函数如下所示：

最大完工时间函数：

$$C_{max} = \max(C_i | i = 1,2,\cdots n) \tag{1}$$

最大机器负荷函数：

$$\mathrm{W_l} = \max(\textstyle\sum_{i=1}^{n}\sum_{j=1}^{n_i} x_{ijk} P_{ijk}) \tag{2}$$

机器总负荷函数：

$$\mathrm{W_{al}} = \textstyle\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i}(x_{ijk} P_{ijk}) \tag{3}$$

2　混合人工蜂群算法

2.1　种群的编码

在 HTABC 算法中，每一个解均由两部分组成：工件序列和机器序列。在工件序列中，包含 n 个工件，每个工件包含若干个工序。对于机器序列，其长度和工件序列相同，每个工序可以由若干个机器生产，通过一定的方式选择一个机器进行生产。工件序列中的每一维代表工件的某一工序，机器序列中的每一维代表对应工序的所选择的机器。

2.2　种群初始化

该阶段需要设定一些参数，主要包括：算法迭代次数，食物源的数目(SN)，雇佣蜂、跟随蜂、侦查峰的数目，食物源不发生变化必须放弃所需要达到的次数($limits$)。对食物源进行初始化，令 $X_i=\{x_{i1}, x_{i2}, x_{i3} \ldots x_{in}\}$为第 i 个食物源，采用公式 4 产生。然后将食物源逐一分配给雇佣蜂。

对于元启发式算法来说，操作序列的好坏直接影响到算法的收敛速度和求解结果，工件序列主要通过随机排列产生，机器序列主要采用随机规则和混合规则相结合的方法产生。如果工序可选择的机器数量不多，那么机器序列可以采用随机规则产生，否则就使用混合规则：首先按时间对该序列机器进行排序，选中操作时间最短的三个机器，然后使用随机方法选择一个机器。混合规则的使用可以使序列得到大大的优化，最大完工时间得到极大的缩短。

2.3　雇佣蜂阶段

雇佣蜂首先会携带着食物源出发，在附近进行搜寻，新食物源生成将使用开发搜索。其中，工件序列和机器序列的交叉操作将分开执行，采用不同的操作方法。具体的操作步骤如下：

a)从种群中挑选三个最优秀的解组成最优解集;

b)种群中所有的解 X_i 将依次执行交叉操作，交叉的对象为最优解集中随机选择的一个解，从而产生新解 X_{new}。该阶段机器序列保持不变，仅仅交叉工件序列;

c)根据机器序列的长度，从机器序列中适当挑选几个工序，重新选择加工机器;

d)将新解 X_{new} 与旧解 X_i 进行对比，如果新解更优，则 X_{new} 将取代 X_i 的位置，否则保持不变。

2.4　跟随蜂阶段

当雇佣蜂返回巢穴后，跟随蜂将选择一个蜜蜂跟随。然后对该区域进行更加深入的搜寻。跟随蜂的操作有助于提高搜索精度与全局收敛性。在解的更新上，将采用小幅度更新策略。以下为具体的操作步骤：

a)为每个跟随蜂选择一个雇佣蜂跟随;

b)对跟随蜂携带的解进行更新操作，工件序列和机器序列的更新幅度将不会太大;

c)判断新解 X_{new} 是否比旧解 X_i 更加优秀，如果是，则新解 X_{new} 代替旧解 X_i，否则保持不变。

2.5　侦查蜂阶段

随着迭代的持续进行，食物源发生变化的几率也逐渐减少。优良的基因逐渐的被最好的食物源(X_{best})所继承，且几乎不会发生变化，为了提高种群的多样性，必须对蜜蜂所携带的食物源限定次数。如果食物源在指定次数内不发生变化，那么该食物源的携带者就会转换为侦查蜂，食物源会被重新初始化。

2.6　禁忌搜索

禁忌搜索算法是一种智能启发式算法，该算法具有灵活的记忆功能，搜索过程中可以接受劣解，从而可以跳出局部最优解。其藐视准则可以接受差解，从而扩大了解的局部搜索能力。禁忌搜索的基本原理是给一个初始解，通过此初始解产生相关的邻域解，从邻域解里选一个最优解作为当前解，如此进行循环直到满足收敛条件。为了避免重复操作，需要设定一个禁忌表，如果当前解在禁忌表里，那么就放弃该解，从邻域里重新选择一个。如果该解特别有益，可以忽略禁忌特性，这就是藐视准则。具体的操作步骤如下所示：

a)选择部分解进行搜索集合。

b)依次从搜索集合中挑选解作为当前解。

c)对当前解产生一定数量的邻域解，选择最优解 y 作为候选解。

d)如果 y 满足藐视准则，将 y 作为当前解并置于禁忌表中；否则从候选解中选择一个满足非禁忌特性的解作为当前解，并置于禁忌表中。

e)释放满足禁忌长度的解。

f)输出新解。

3　实例验证

本章主要内容是测试 HTABC 算法在求解高质量解方面的表现，算法采用 MATLAB 编程。算法的有效性方面的验证将采用 Kacem 算例，该算例包括五个实例，分别为 4×5，8×8，10×7，10×10，15×10。实验所取的值是 10 次独立运行后的最优解。

HTABC 算法的相关参数设置如下：工件的数目 n，机器的数目 m，操作总数目 $toper$，迭代次数 $iter_max$=200，种群数目 PN=40，$limit$ 数目为 10，禁忌搜索候选解个数为 5，禁忌搜索最大迭代次数 2*n，禁忌表长度为 $toper$/2。

为了验证算法优越的性能，比较的对象均是 FJSP 领域采用不同算法求解的最优算法，包括改进遗传算法(IGA)[1]，粒子群算法与禁忌搜索相结合的混合算法(PSO+TS)[8]，基于 Pareto 的混合人工蜂群算法(P-DABC)[10]，分布估计算法(EDA)[14]，混合人工蜂群算法(DABC)[15]。比较结果如表 1 所示，其中每个实例均由三行组成，从上至下分别为最大完工时间、最大的机器负荷、机器总负荷。如果某算法关于某一个实例没有数据，则用符号“N/A”表示。

表 1　5 个 Kacem 算例的比较结果

	PSO+TS		IGA	P-DABC			EDA		DABC			HTABC		
	11		N/A	11	12	13	11	11	11	12	11	11	11	11
4×5	10		N/A	10	8	7	10	9	10	8	9	10	9	10
	32		N/A	32	32	32	32	34	32	32	34	32	34	33
	14	15	14	14	15	16	14	15	14	15	16	14	15	15
8×8	12	12	12	12	12	13	12	12	12	12	13	12	15	12
	77	75	77	77	75	73	77	75	77	75	73	77	75	75
	N/A		N/A	11	12	12	11	11	11	11	12	11	11	11
10×7	N/A		N/A	11	11	12	11	10	11	10	12	11	11	11
	N/A		N/A	63	61	60	61	62	61	62	60	61	61	61
	7		7	7	8	8	7	7	7	8	7	7	7	7
10×10	6		6	5	7	5	6	5	6	5	5	6	7	6
	43		42	43	41	42	44	43	42	42	43	41	41	42
	11		11	11	12		11	11	11	11		11	12	12
15×10	11		11	11	11		10	11	10	11		11	11	12
	93		91	93	91		93	91	93	91		93	91	96

由表 2 可知，对于实例 4×5 和实例 8×8，通过和其他几个算法作比较可以发现包括最大完工时间在内的三个指标均达到其他算法最好的状态。对于实例 10×7，通过比较可以发现，HTABC 算法和 EDA 及 DABC 具有同样的最优值，且比 P-DABC 稍好一些。本算法求解实例 10×7 所获得的最优解为：Cmax=11, Wl = 11, Wal = 61，其甘特图如图 1 所示。其中每一个方块代表一个工序,方块内会有工件号和工序号，方块对应的纵坐标是操作机器，方块左边竖线代表工件开始时间，右边竖线代表结束时间。

对于实例 10×10，通过比较发现，HTABC 算法和其他算法的 C_{max} 均能达到最优值，但是在 C_{max} 一定的情况下，本算法的机器总负荷更小。实例 15×10 是 Kacem 问题最复杂的一个，通

过与其他算法作比较，HTABC 除了稍逊于 EDA, DABC 之外，和 PSO+TS, P-DABC 拥有同样的最优值，且比 PSO+SA 稍优。本算法求解实例 15×10 所获得的最优解为：C_{max}=11，W_l = 11，W_{al} =93，其甘特图如图 2 所示。

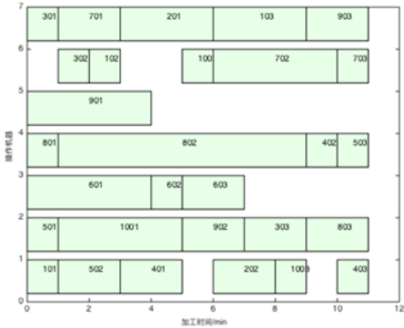


图 1　实例 10×7 最优解

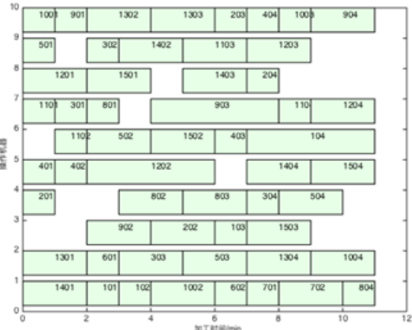


图 2　实例 15×10 最优解

4　结束语

本文通过使用基于禁忌搜索 HTABC 算法通过建立三个目标求解 FJSP。通过将基于有效的种群初始化策略的 HTABC 算法和禁忌搜索策略相结合的混合算法可以很好地在整个解空间搜索到最优解。在种群初始化阶段采用多策略相结合方法。在雇佣蜂阶段采用开发搜索与最优解进行交叉产生新解。跟随蜂阶段对工件序列和机器序列采用小幅度更新策略，使得最优解得到进一步优化。禁忌搜索的应用提高了解局部搜索能力。算法使用 Kacem 算例进行测试，与当前采用不同算法解决多目标 FPSP 最有竞争力的实例作对比。通过比较发现算法的每个实例均能达到 Kacem 算例最优值。实验结果和比较证明了 HTABC 算法的在求解多目标 FJSP 上的有效性和高效性。

算法的不足之处在于 HTABC 算法关于 Kacem 算例的数据需要进一步优化。在未来将进一步去提高算法的性能，尝试将 HTABC 算法用于其他生产调度问题，考虑基于工件插入问题的 FJSP。

参考文献：

[1]. 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题[J]. 机械工程学报, 2009, 45(7):145-151.

[2]. 张国辉, 党世杰. 考虑工件移动时间的柔性作业车间调度问题研究[J]. 计算机应用研究, 2017, 34(8):2329-2331.

[3]. Wang S, Yu J. An effective heuristic for flexible job-shop scheduling problem with maintenance activities [J]. Computers & Industrial Engineering, 2010, 59(3):436-447.

[4]. 苏子林, 车忠志, 苑金梁. 柔性作业车间调度问题的一种启发式算法[J]. 计算机应用研究, 2011, 28(6):2060-2063.

[5]. Shao X, Liu W, Liu Q, et al. Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem [J]. International Journal of Advanced Manufacturing Technology, 2013, 67(9):2885-2901.

[6]. Zhang C, Wang X, Gao L. An improved genetic algorithm for multi-objective flexible job-shop scheduling problem [J]. Advanced Materials Research, 2010, 97(101):2449-2454.

[7]. 马雪丽, 曹德弼, 刘晓冰. 面向柔性工艺的作业车间调度问题混合遗传算法[J]. 计算机应用研究, 2014, 31(5):1353-1357.

[8]. Zhang G, Shao X, Li P, et al. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem [J]. Computers & Industrial Engineering, 2009, 56(4):1309-1318.

[9]. 吴秀丽, 张志强, 杜彦华, 等. 改进细菌觅食算法求解柔性作业车间调度问题[J]. 计算机集成制造系统, 2015, 21(5):1262-1270.

[10]. Li J, Pan Q, Gao K. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems [J]. International Journal of Advanced Manufacturing Technology, 2011, 55(9):1159-1169.

[11]. Thammano A, Phu-Ang A. A hybrid artificial bee colony algorithm with local search for flexible Job-Shop scheduling problem [J]. Procedia Computer Science, 2013, 20:96-101.

[12]. Banharnsakun A, Achalakul T, Sirinaovakul B. The best-so-far selection in artificial bee colony algorithm [J]. Applied Soft Computing, 2011, 11(2):2888-2901.

[13]. Wang L, Zhou G, Xu Y, et al. An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible Job-Shop scheduling [J]. International Journal of Advanced Manufacturing Technology, 2012, 60(9):1111-1123.

[14]. Wang S, Wang L, Liu M, et al. An effective estimation of distribution algorithm for the flexible Job-Shop scheduling problem with fuzzy processing time [J]. International Journal of Production Research, 2013, 51(12):3778-3793.

[15]. Li J, Pan Q, Tasgetiren M F. A discrete artificial bee colony algorithm for the multi-objective flexible Job-Shop scheduling problem with maintenance activities [J]. Applied Mathematical Modeling, 2014, 38(3):1111–1132.